

SERVER PROGRAM MANAGEMENT METHOD

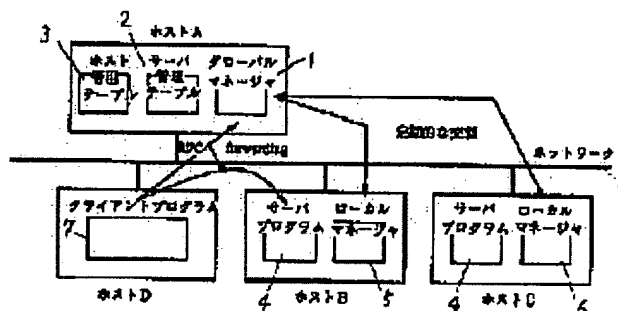
Patent number: JP6266643
Publication date: 1994-09-22
Inventor: WADA HIDEHIKO
Applicant: YOKOGAWA ELECTRIC CORP
Classification:
 - **International:** G06F13/00; G06F15/16; H04L12/40
 - **European:**
Application number: JP19930057005 19930317
Priority number(s): JP19930057005 19930317

[Report a data error here](#)

Abstract of JP6266643

PURPOSE:To call an optimum server program when plural server programs of a remote procedure call exist on a network.

CONSTITUTION:Local managers 5 and 6 transmit the performance of host computers B and C to which they themselves belong to a global manager 1, and transmit a load average to the global manager 1 at a prescribed period. The global manager 1 initializes management tables 2 and 3, and waits for communication from the local managers 5 and 6 or a client program 7. In the case of communication from the local managers 5 and 6, information of the host computer where the local manager concerned exists is received and priority is obtained based on the information. In the case of communication from the client program 7, a message is received, the host computers where the server program of the remote procedure call exists are searched, the host computer of the highest priority is selected and the message is forwarded.



Data supplied from the [esp@cenet](#) database - Worldwide

This Page Blank (uspio)

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-266643

(43)公開日 平成6年(1994)9月22日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 13/00	3 5 7 Z	7368-5B		
15/16	3 7 0 Z	7429-5L		
H 0 4 L 12/40		7341-5K	H 0 4 L 11/ 00	3 2 0

審査請求 未請求 請求項の数1 O L (全 8 頁)

(21)出願番号 特願平5-57005

(22)出願日 平成5年(1993)3月17日

(71)出願人 000006507

横河電機株式会社

東京都武蔵野市中町2丁目9番32号

(72)発明者 和田 英彦

東京都武蔵野市中町2丁目9番32号 横河
電機株式会社内

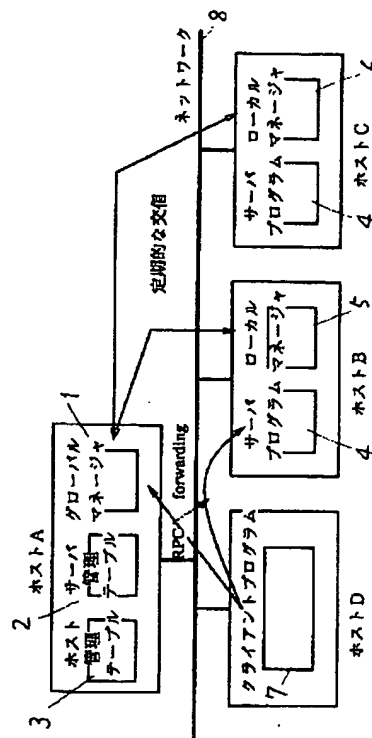
(74)代理人 弁理士 小沢 信助

(54)【発明の名称】 サーバプログラム管理方法

(57)【要約】

【目的】 ネットワーク上にリモートプロシージャコールのサーバプログラムが複数個存在する時に、最適なサーバプログラムを呼び出すことのできる管理方法を提供する。

【構成】 ローカルマネージャは自身が属するホストコンピュータの性能を前記グローバルマネージャへ送信し、その後は一定周期でロードアベレージをグローバルマネージャへ送信し、グローバルマネージャは、前記管理テーブルを初期化した後前記ローカルマネージャまたはクライアントプログラムからの通信を待ち、①ローカルマネージャからの通信のときは、当該ローカルマネージャの存在するホストコンピュータの情報を受け取り、その情報に基づき優先度を求め、②クライアントプログラムから通信のあったときは、メッセージを受信し、リモートプロシージャコールサーバプログラムの存在するホストコンピュータを探し、そのうちで最も優先度の高いホストコンピュータを選択し前記メッセージをフォワードする。



【特許請求の範囲】

【請求項1】複数のホストコンピュータがネットワークを通して接続され、リモートプロシージャコールサーバプログラムが複数のホストコンピュータに存在するシステムにおいて、

あるホストコンピュータ（A）上に、ネットワーク全体のリモートプロシージャコールサーバプログラムおよびこのサーバプログラム（4）が動作するホストコンピュータに関する情報を管理テーブルに記憶し、他のホストコンピュータ（D）のクライアントプログラム（7）からのリクエストを受けるとサーバホストコンピュータ

（B、C）の性能や負荷に基づき適切なホストコンピュータを選択し、当該ホストコンピュータのサーバプログラム（4）へメッセージをフォワードするグローバルマネージャ（1）を備え、

前記サーバプログラム（4）の存在するホストコンピュータ上に、サーバプログラムおよび自身のホストコンピュータの情報を前記グローバルマネージャ（1）に送信するローカルマネージャ（5、6）を備え、

このローカルマネージャ（5、6）は自身が属するホストコンピュータの性能を前記グローバルマネージャ

（1）へ送信し、その後は一定周期でロードアベレージをグローバルマネージャ（1）へ送信し、

前記グローバルマネージャ（1）は、前記管理テーブルを初期化した後前記ローカルマネージャ（5、6）またはクライアントプログラム（7）からの通信を待ち、

(a) ローカルマネージャ（5、6）から通信のあったときは、

当該ローカルマネージャの存在するホストコンピュータの情報を受け取り、

その情報に基づき定められた関係で優先度を求め、

(b) クライアントプログラム（7）から通信のあったときは、

メッセージを受信し、リモートプロシージャコールサーバプログラム（4）の存在するホストコンピュータを探し、そのうちで最も優先度の高いホストコンピュータを選択しそのホストコンピュータに前記メッセージをフォワードするようにして、リモートプロシージャコールサーバプログラムを呼び出すことができるようにしたことを特徴とするサーバプログラム管理方法。

【発明の詳細な説明】**【0001】**

【産業上の利用分野】本発明は、ネットワーク上に複数のホストが接続された分散コンピューティング環境において、複数のホストコンピュータ上でリモートプロシージャコール（RPC）のサーバプログラムが動作している場合に、リモートプロシージャコールのクライアントプログラムが最適なホストのサーバプログラムへの呼び出しを行なうことができるようにするサーバプログラム

【0002】

【従来の技術】リモートプロシージャコールは、図11に示すようにホストコンピュータ（以下ホストという）A上のプログラム（クライアントプログラム）11が異なるホストB上のプログラム（サーバプログラム）12を呼び出し、その結果（リターンバリューという）を返してもらうというものである。この時クライアントプログラムは、サーバプログラムに対してプログラム名とバージョン番号を含んだメッセージを送る。

【0003】図12に示すようにネットワークで接続されたホストの台数が増えてくると、同じ機能のサーバプログラムが複数のホスト上で動作する場合がある。この時にどのホストのサーバプログラムを選択するかということがリモートプロシージャコールの効率に大きく影響する。従来の、リモートプロシージャコールのサーバプログラム管理方法として代表的なものは、

①ローカルホストのサーバプログラムだけを管理する方法、

②ネットワーク内のサーバプログラムを管理する方法である。

【0004】

【発明が解決しようとする課題】しかしながら、上記①の方法ではネットワーク全体の管理ができず、また②の方法では異なるホスト上で動く同じ機能のサーバプログラムを効率を含めて管理することができないという問題があった。

【0005】本発明はこのような点に鑑み、ネットワーク上にリモートプロシージャコールのサーバプログラムが複数個存在する時に、クライアントプログラムがサーバプログラムから効率よくリターンバリューを返してもらうために、一番適切なサーバプログラムを呼び出すことのできる管理方法を提供することを目的とする。

【0006】

【課題を解決するための手段】このような目的を達成するために本発明では、複数のホストコンピュータがネットワークを通して接続され、リモートプロシージャコールサーバプログラムが複数のホストコンピュータに存在するシステムにおいて、あるホストコンピュータ上に、ネットワーク全体のリモートプロシージャコールサーバプログラムおよびこのサーバプログラムが動作するホストコンピュータに関する情報を管理テーブルに記憶し、他のホストコンピュータにあるクライアントプログラムからのリクエストを受けるとサーバホストコンピュータの性能や負荷に基づき適切なホストコンピュータを選択し、当該ホストコンピュータのサーバプログラムへメッセージをフォワードするグローバルマネージャを備え、前記サーバプログラム（4）の存在するホストコンピュータ上に、サーバプログラムおよび自身のホストコンピ

ーカルマネージャを備え、

【0007】このローカルマネージャは自身が属するホストコンピュータの性能を前記グローバルマネージャへ送信し、その後は一定周期でロードアベレージをグローバルマネージャへ送信し、前記グローバルマネージャは、前記管理テーブルを初期化した後前記ローカルマネージャまたはクライアントプログラムからの通信を待ち、(a) ローカルマネージャから通信のあったときは、当該ローカルマネージャの存在するホストコンピュータの情報を受け取り、その情報に基づき定められた関係で優先度を求め、(b) クライアントプログラムから通信のあったときは、メッセージを受信し、リモートプロシージャコールサーバプログラムの存在するホストコンピュータを探し、そのうちで最も優先度の高いホストコンピュータを選択し前記メッセージをフォワードするようにして、リモートプロシージャコールサーバプログラムを呼び出すことができるようにしたことを特徴とする。

【0008】

【作用】サーバプログラムが存在するホストコンピュータ(B、C)のローカルマネージャからは、自身のホストコンピュータの性能を、グローバルマネージャの存在するホストコンピュータ(A)へ送信すると共に、ホストコンピュータ(B、C)のロードアベレージを一定周期ごとに送信する。ホストコンピュータ(A)では、初期化の後通信待ちに入り、ローカルマネージャからの通信の場合は、当該ホストコンピュータの性能やロードアベレージを受け取って管理テーブルに格納し、管理テーブルを参照してホストコンピュータの優先度を求めておく。クライアントプログラムからの通信の場合は、リモートプロシージャコール対象のサーバプログラムのあるホストコンピュータの内で最も優先度の高いホストコンピュータを選択し、クライアントプログラムからのメッセージを選択されたホストコンピュータのポートへ送る。このような管理により、リモートプロシージャコールにおいて最適なサーバプログラムを選択することができる。

【0009】

【実施例】以下図面を用いて本発明を詳しく説明する。図1は本発明の管理方法を実施するためのシステムの一例を示す要部構成図である。図において、ホストA、B、C、Dはネットワーク8でそれぞれ接続されており、ホストBとC上で同じ機能のサーバプログラムが動作しているものとする。また、ホストDではクライアントプログラムが動作するものとする。

【0010】ホストAは、グローバルマネージャ1、サーバプログラム管理テーブル2、ホスト管理テーブル3を有する。グローバルマネージャ1は、①ホストBやCのローカルマネージャ(詳細は後述)と定期的に通信し、サーバプログラムおよびサーバプログラムが動作す

行なう機能、②サーバプログラム管理テーブルの管理を行なう機能、③クライアントからのリクエストを受け付け、適切なサーバが動作するホストへリクエストをフォワーディングする機能を有する。

【0011】サーバプログラム管理テーブル2は、グローバルマネージャ1が動作するホスト上のメモリ(図示せず)に存在し、グローバルマネージャ1がローカルマネージャと通信をして得られた、ネットワーク上で動作するリモートプロシージャコールサーバプログラムの情報を保持しておく領域である。このテーブルは例えば図2に示すような形式になっており、グローバルマネージャ1によって管理される。

【0012】ホスト管理テーブル3は、グローバルマネージャ1が動作するホストAのメモリに存在し、グローバルマネージャ1がローカルマネージャと通信をして得られた各ホストの情報[すなわち処理能力(性能とも言う)とロードアベレージ(負荷とも言う)]を保持しておく領域であり、グローバルマネージャ1が管理する。このホスト管理テーブル3には例えば図3のように、ホストエントリ番号、ホストアドレス、性能、ロードアベレージ、優先度に関する情報が登録される。

【0013】ホストB、Cはローカルマネージャ5(6)とサーバプログラム4をそれぞれ有し、ホストDはクライアントプログラム7を有す。ローカルマネージャ5(6)は、リモートプロシージャコールのサーバプログラム4が動作するホスト上にあり、サーバプログラムやホストの情報(処理能力、ロードアベレージ)を定期的にグローバルマネージャ1に送信する。

【0014】なお、サーバプログラム4は、ホストB、Cのメモリ(図示せず)上でそれぞれ動作する同機能のプログラムであり、クライアントプログラム7からの呼び出しに応じて計算結果を返すようになっている。なお、サーバプログラム4は起動時にローカルマネージャ5(6)に登録される。クライアントプログラム7は、ホストDのメモリ(図示せず)上で動作し、ホストAのグローバルマネージャ1を呼び出すことによって他ホストで動作するサーバプログラムを呼び出す。

【0015】次に、このようなシステムにおける動作について説明する。なおここでは、ホストAのグローバルマネージャ1がどのようにして最適なサーバプログラムを決定するかに注目し、それに関連する動作を説明する。

【0016】グローバルマネージャ1は、図4に示すように、初期化を行った後、ローカルマネージャ5(6)あるいはクライアントプログラム7からの通信を待つ。クライアントプログラムからの通信であれば、クライアントプログラムの通信に関する処理を行う。その処理が終了すれば、再び元のローカルマネージャとクライアントプログラムからの通信待ちに移る。他方ローカルマネ

らの通信に関する処理を行なう。終了後は、再び元のローカルマネージャとクライアントプログラムからの通信待ちに移る。

【0017】次に上記各処理、すなわち初期化、クライアントプログラムからの通信に関する処理、ローカルマネージャからの通信に関する処理についてそれぞれ説明する。まず初めに前記初期化処理について説明する。初期化処理とは、図5に示すようにサーバプログラム管理

テーブル2およびホスト管理テーブル3をそれぞれ初期化し、その後ローカルマネージャあるいはクライアントプログラムからのメッセージ待ちの状態に入る処理である。

【0018】クライアントプログラムの通信に関する処理とは、図6に示すように、クライアントプログラムからのメッセージを受信し、クライアントプログラムのプログラム名からサーバプログラム管理テーブル2（図2）の該当エントリを探し、エントリが見つけれらる

と、そのエントリにおけるホストエントリ番号とポート番号のリスト（ポインタによりリンクされている）に基づいてホストプログラム管理テーブル3（図3）を参照し、優先度の最も高いホストを選択する。選択されたホスト番号はサーバプログラム管理テーブル2の最終呼び出しホスト番号欄に書き込まれる。このようにして選択されたホストに対してそのポートへクライアントプログラムからの前記メッセージをフォワードする。

【0019】ローカルマネージャからの通信に関する処理とは、グローバルマネージャ1のローカルマネージャに関する処理であって、図7に示すように、ホスト情報（性能）を受信した場合にはそのホスト情報をホスト管理テーブル3へ登録する。この登録終了後あるいは初めにホスト情報を受信しなかった場合には、次のロードアベレージ受信に移る。ロードアベレージが受信されるとそれをホスト管理テーブル3へ登録する。登録終了後あるいはロードアベレージを受信しなかった場合には、続いて優先度の再計算を行なう。優先度の計算は次の式に従う。各ホストの性能を X （ $0 \leq X \leq 10$ ）、ロードアベレージを Y （ $0 \leq Y \leq 10$ ）とすると、優先度 P は、 $P = \omega_x X + \omega_y Y$

で与えられる。ただし、 $\omega_x + \omega_y = 1$ であり、 ω_x 、 ω_y の値は性能とロードアベレージのどちらに比重を置くかを示しており、適宜に設定される。また、ローカルマネージャ5（6）は、立ち上げ時に処理能力 X の値をグローバルマネージャ1に送信し、また定期的にロードアベレージ Y の値をグローバルマネージャ1に送信する。

【0020】次にローカルマネージャ5（6）の動作について説明する。ローカルマネージャ5（6）は、図8に示すように、初期化と、グローバルマネージャ1へのホスト情報の送信処理を行なった後、ロードアベレージ

る動作を繰り返す（この繰り返し部分の動作を定周期動作という）。

【0021】なお、ここでの初期化は、図9に示すように、グローバルマネージャ1へ接続し、接続に成功した場合はホスト情報の1つである処理能力を送信し、その後サーバプログラムの登録受け待ちの状態になる。前記接続に失敗した場合はいきなりサーバプログラムの登録受け待ちの状態に移るようにした処理である。

【0022】また、前記定周期動作は、図10に示すように、グローバルマネージャ1への接続を試みる。接続に成功すると、グローバルマネージャ1とは既に接続済であるかどうかを調べる。接続済でない場合にはホスト情報の性能の送信とロードアベレージ（負荷）の送信を行なう。接続済の場合はロードアベレージの送信のみ行なう。上記ロードアベレージの送信が終了すると、その後は一定時間待ちの処理に入る。なお最初のグローバルマネージャ1への接続において失敗した場合も、この一定時間待ちの処理に入る。

【0023】要するに本発明の管理方法においては、ホストDのクライアントプログラムからリモートプロシージャのリクエストがあると、ホストAのグローバルマネージャ1はサーバホストB、Cの性能や負荷を考慮して最適なサーバプログラムを見つけ出し、そのサーバプログラムへメッセージをフォワードする。他方、各ホスト（ホストB、C）のローカルマネージャからは、そのホスト内のリモートプロシージャコールサーバプログラムの情報や、自身のホストの性能および負荷情報をグローバルマネージャに送る。

【0024】なお、実施例ではホストコンピュータが4個ネットワーク接続した場合について示したが、本発明はこれに限定されるものではない。また、グローバルマネージャと管理テーブル、ローカルマネージャ5とサーバプログラム4などはどのホストコンピュータ上にあっても差し支えない。

【0025】

【発明の効果】以上説明したように本発明によれば、ネットワーク上の複数のホスト上で動作するリモートプロシージャコールのサーバプログラムを1つのグローバルマネージャで管理することができ、クライアントプログラムからのリクエストがあると、効率を考慮して一番適切なホストへ割り振ることが可能となる。

【図面の簡単な説明】

【図1】本発明に係るサーバプログラムの管理方法を実現するための装置の一実施例構成図である。

【図2】サーバプログラム管理テーブルの一例を示す図である。

【図3】ホストプログラム管理テーブルの一例を示す図である。

【図4】グローバルマネージャの動作フローを示す図で

【図5】グローバルマネージャの初期化部分の動作フローを示す図である。

【図6】クライアントプログラムからの通信に関する処理に係る動作フローを示す図である。

【図7】ローカルマネージャからの通信に関する処理に係る動作フローを示す図である。

【図8】ローカルマネージャの動作フローを示す図である。

【図9】ローカルマネージャの初期化部分の動作フローを示す図である。

【図10】ローカルマネージャの定周期動作に係る動作フローを示す図である。

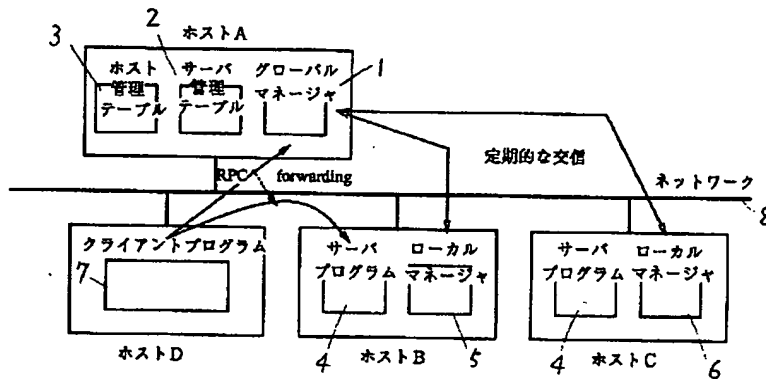
【図11】リモートプロシージャコールを説明するための説明図である。

【図12】複数ホスト上でのサーバの動作を説明するための説明図である。

【符号の説明】

- 1 グローバルマネージャ
- 2 サーバプログラム管理テーブル
- 3 ホスト管理テーブル
- 4 サーバプログラム
- 5、6 ローカルマネージャ
- 7 クライアントプログラム
- A, B, C, D ホストコンピュータ

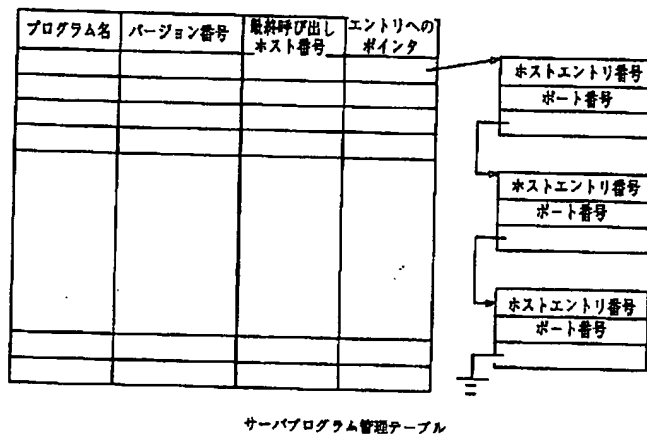
【図1】



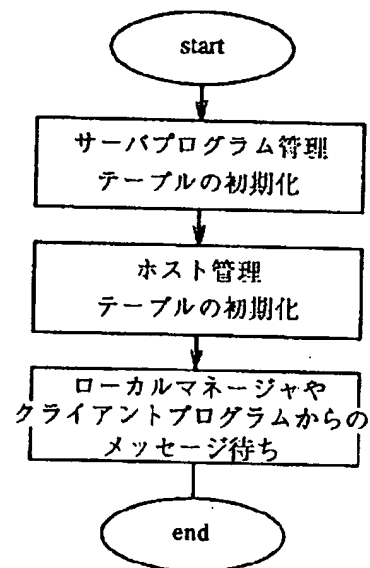
【図3】

ホストエントリ番号	ホストアドレス	性能	ロードアベレージ	優先度

【図2】

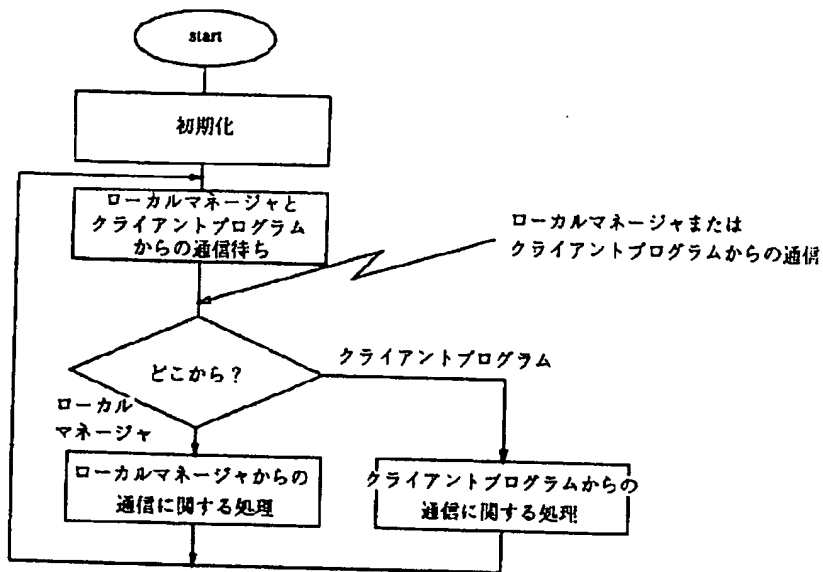


【図5】



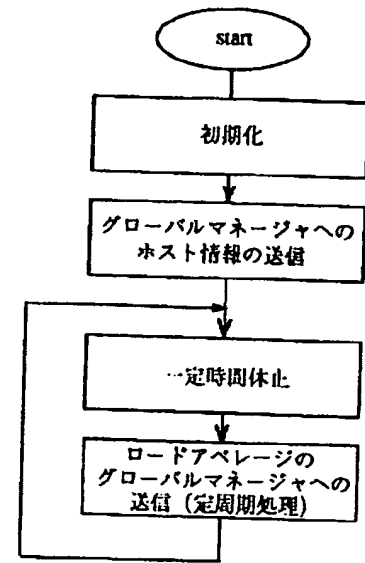
グローバルマネージャの動作（初期化部分）

【図4】



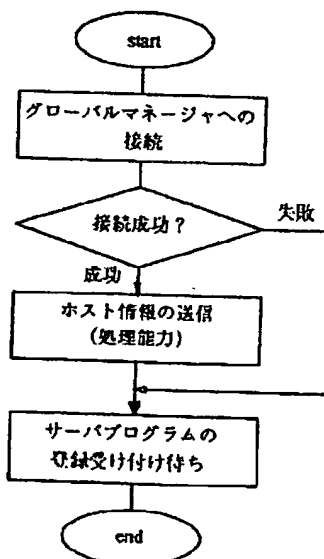
グローバルマネージャの動作

【図8】



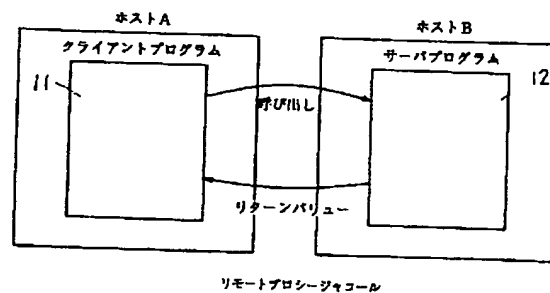
ローカルマネージャの動作

【図9】

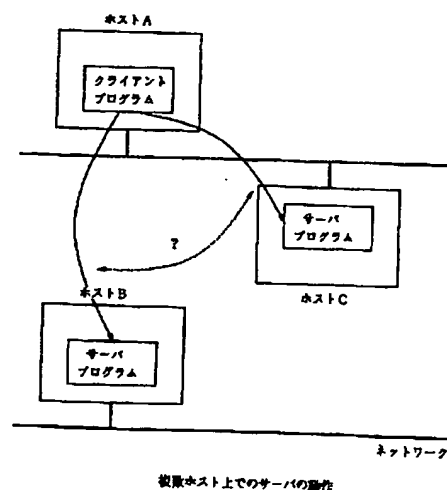


ローカルマネージャの動作(初期化部分)

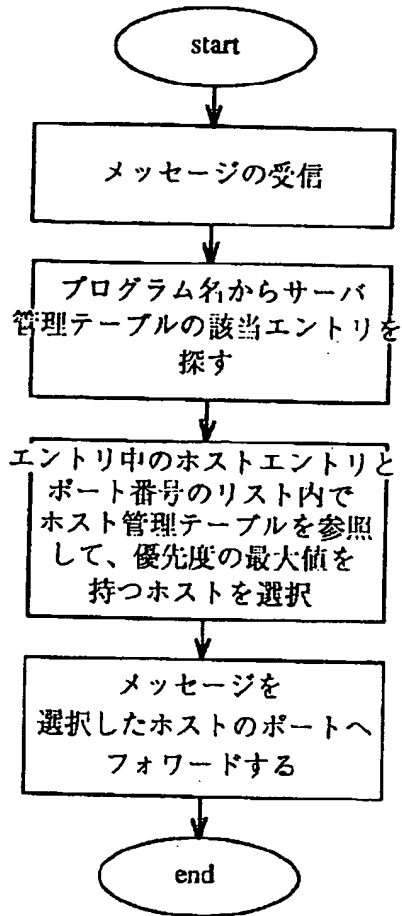
【図11】



【図12】

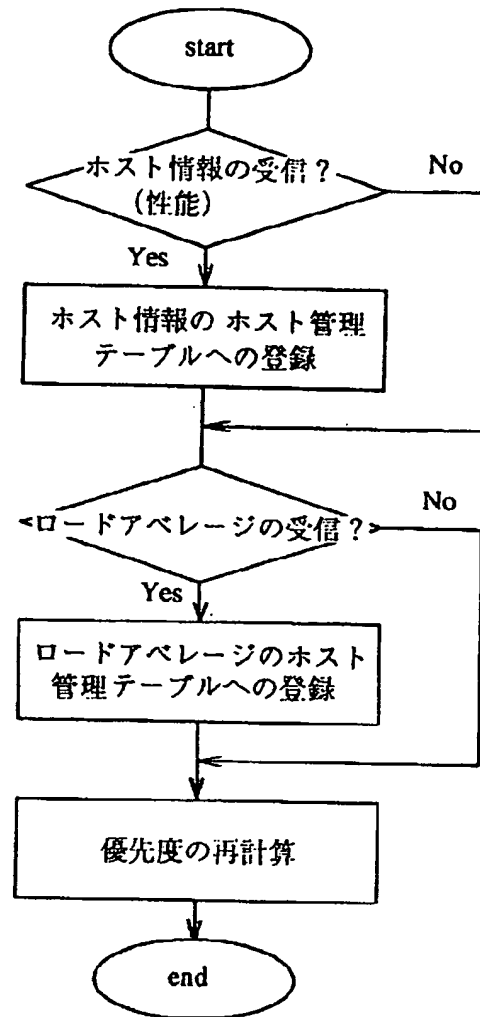


【図6】



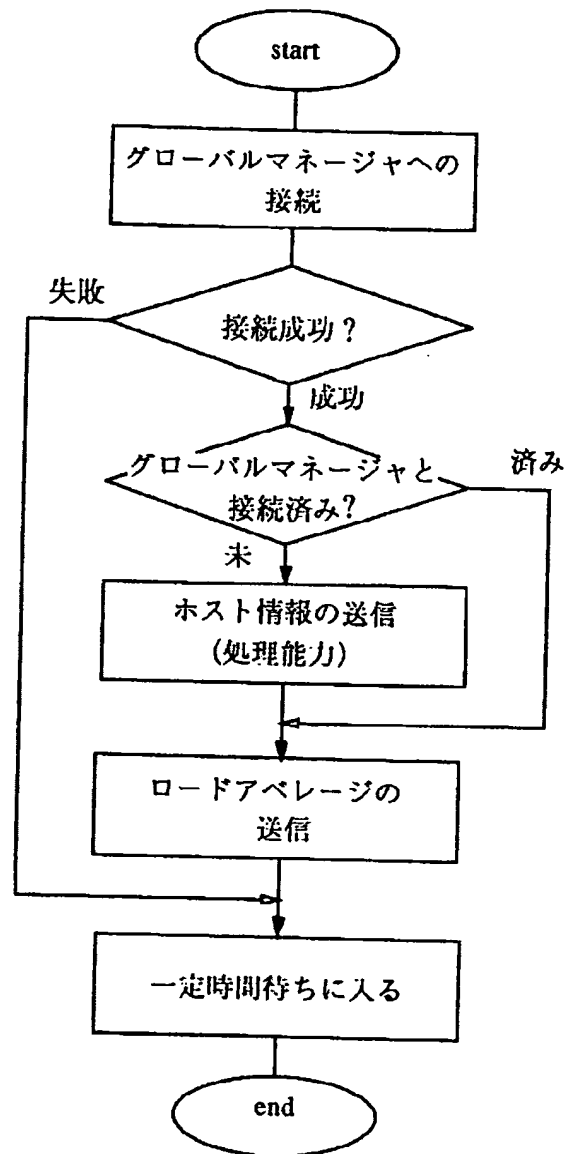
グローバルマネージャの対クライアントプログラム処理

【図7】



グローバルマネージャの対ローカルマネージャ処理

【図10】



ローカルマネージャの動作 (定周期動作)